Goldilocks User Manual

18 August 2013

- Introduction
- Arduino Installation
- Goldilocks Hardware
  - ATmega1284p Features
  - ATmega32U2 Features
  - Layout / Header Pins
  - Solder Bridge Descriptions
    - I2C Arduino Pre-R3 Support
    - SPI Interconnect
    - Rx Tx Disconnect
    - ATmega32U2 Rx Tx LED Disconnect
    - ATmega32U2 HWB (DFU Bootloader) & ATmega1284p RESET Disconnect
  - Selected Feature Descriptions
    - Prototyping space
    - uSD Card
    - ATmega1284p JTAG Header
    - 32.768kHz Crystal
    - RESET buttons
    - 32U2 Port Header
    - AVR Fuses
    - Power Configuration
- Arduino Compatibility
  - Features
  - Details
  - Known Arduino R3 Incompatibilities
- ATmega1284p Firmware
  - Wiring (stk500v2) ATmega1284p Bootloader
- ATmega32U2 Firmware
  - LUFA DFU Bootloader
  - LUFA Benito (U2duino) Serial Interface
- Schematic
- Errata
- Support
  - Contact Information
  - Support Information
  - Precautions
- Legal
  - Copyright Notices
  - Software Warranty Disclaimer

Hardware Disclaimer
Trademarks
Acknowledgments

Introduction

The Goldilocks is an advanced AVR development PCB loaded with both the ATmega1284p microcontroller and the ATmega32U2 USB microcontroller. Goldilocks is intended to conform to the Arduino Uno or Leonardo physical format and interface standards, bringing significant improvements and longevity into the existing Arduino AVR platform, whilst providing full compatibility with both R3 and pre-R3 Arduino Shields.

Whilst the Goldilocks is easy to use and is designed and built with care, it is not intended to be an entry point into the world of Arduino and microcontrollers in general. The easiest entry point remains the standard Arduino Uno R3, or cloned device. When you find you are being constrained by what your Arduino can do, you should come back to love the Goldilocks.

The Goldilocks implements many features intended to increase the flexibility of the Arduino platform, allowing advanced projects to be completed, but retaining full compatibility for Arduino Shields to give users access to the full portfolio of options developed for the Arduino Uno and its peers and clones. These features, listed below, will be covered in detail in later sections of this manual.

- Through hole connectors for all I/O pins aligned to 0.1” pitch for breadboard or ribbon-cable compatibility.
- Prototyping space, with stitched cross connections, aligned to 0.1” pitch, and extended outside standard Arduino Uno shield profile.
- Both ATmega1284p and ATmega32U2 ports pinout available in pin logical order.
- Extra 2 x Analogue and 2 x Digital I/O pins.
- I2C pins replicated near to the prototyping space.
- I2C interconnect option for Pre-R3 shield compatibility.
- LED indicators located on end outside standard shield profile for unimpeded visibility.
- ATmega1284p clocked at 20 MHz using precision crystal, with option of 22.1184 MHz.
- 32.768 kHz clock crystal for accurate real-time clock timekeeping.
- Independent Analogue ADC Platform, with filtered AVcc supply.
- AREF replicated near Analogue pins for easy off-board usage.
- Access to 3x programmable USART (2x ATmega1284p & 1x ATmega32U2).
- Serial USART disconnect option.
- Independent MCU reset buttons (ATmega1284p and ATmega32U2).
- JTAG (IEEE std. 1149.1 compliant). **SEE ERRATA**
- Improved PWM pin access.
- SPI interconnect option between ATmega1284p and ATmega32U2, with decoupling.
- Micro SD Card cage using additional SPI Slave Select and Card Detect I/O pins to ensure no impact on existing shield compatibility. **SEE ERRATA**
- High capacity 7V to 28V Switch Mode Power Supply, specified for 5V 2A.
- 4 layer PCB with industrial grade ground plane and power supply decoupling practices.

The 4 layer PCB is high quality with ENIG (gold plated) finish, a gold solder mask, and black screenprinting. The Goldilocks can be powered at 7V to 28V via the 2.1mm barrel connector, or at 5V via USB, depending on the position of the POWER Jumper.

Arduino Installation

Unless you are certain of what you are doing, it is recommended to install the Goldilocks separately from your existing Arduino 1.0.x installations, until you're sure everything is working as expected.

1. Download and install Arduino version 1.0.5, or greater, from http://arduino.cc/en/Main/Software
2. Download the Goldilocks file from Goldilocks 20130818 (also see Freetronics Goldilocks product page for future support and firmware).
3. Unzip this file into a user directory (ie: My Documents/Arduino). You may need to create this folder. Do not unzip into the arduino system directory from Step 1.
4. Inside there is a directory arduino_goldilocks/ which contains a hardware/ directory. This directory can be placed into the Arduino installation folder you prepared previously. It will ask if you want to replace the boards.txt file, and programmers.txt file, which you should accept. Other files should all be new and will be stored quietly.
5. Start the Arduino IDE, and note that the Goldilocks 20 MHz can be selected in the Board selection menu.

Arduino version 1.0.5 supports the addition of the boards.txt and other files directly into your sketch folder. Please refer to the quick-start instructions accompanying the Goldilocks packaging for instruction on this method of adding the necessary files.

If you are using Windows, you will need obtain a INF file for the USB Serial interface. As the standard LUFA USB nonclamenture is used, the LUFA U2duino.INF found in the production_firmware folder of the Goldilocks 20130818 file should be installed.

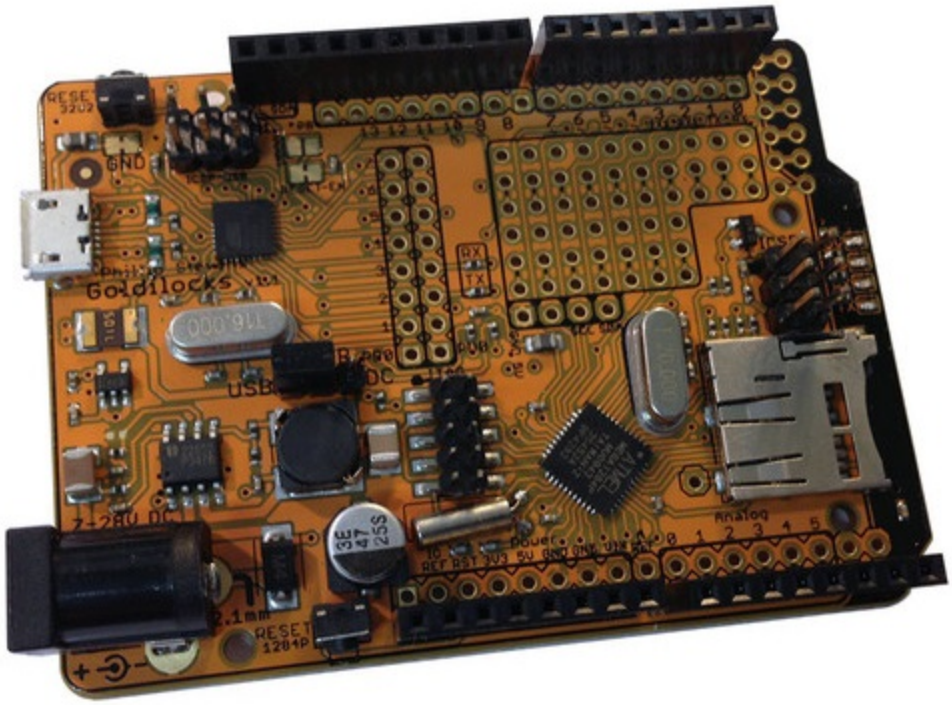
Goldilocks Hardware

ATmega1284p Features

The ATmega1284P provides the following features: 128 kBytes of In-System Programmable Flash with Read-While-Write capabilities, 4 kBytes EEPROM, 16 kBytes SRAM, Real Time Counter (RTC), three flexible Timer/Counters with compare modes and PWM, 2 USARTs, one byte oriented 2-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain, one SPI serial port, IEEE std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six software selectable power saving modes.

ATmega32U2 Features

The ATmega32U2 provides the following features: 32 kBytes of In-System Programmable Flash with Read-While-Write capabilities,1 kByte EEPROM,1 kByte SRAM, two flexible Timer/Counters with compare modes and PWM, one USART, one SPI serial port, debugWIRE interface, also used for accessing the On-chip Debug system and programming and five software selectable power saving modes.



Layout / Header Pins

The below table identifies the layout of the Goldilocks PCB pins with respect to the Arduino Uno R3 board including a comparison of relevant features and provides commentary where relevant.

Arduino UNO R3	328p Feature	328p Pin	1284p Pin	1284p Feature	Goldilocks Comment
Analog 0		PC0	PA0		
Analog 1		PC1	PA1		
Analog 2		PC2	PA2		
Analog 3		PC3	PA3		
Analog 4	SDA	PC4	PA4		Solder bridge to PC1 SDA I2C
Analog 5	SCL	PC5	PA5		Solder bridge to PC0 SCL I2C
			PA6		Analog 6 & Through Hole
			PA7		Analog 7 & Through Hole
Reset	Reset	PC6	RESET		Separate RESET Pin
Digital 0	RX	PD0	PDO	RX0	
Digital 1	TX	PD1	PD1	TX0	
Digital 2	INT0	PD2	PD2	INT0 / RX1	Additional USART1 Rx
Digital 3	INT1 / PWM2	PD3	PD3	INT1 / TX1	Additional USART1 Tx
Digital 4		PD4	PD4	PWM1	16 bit PWM clear of SPI pins **SEE ERRATA**
Digital 5	PWM0	PD5	PD5	PWM1	16 bit PWM clear of SPI pins
Digital 6	PWM0	PD6	PD6	PWM2	
Digital 7		PD7	PD7	PWM2	

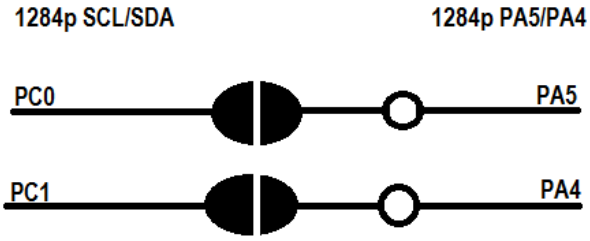
Digital 8		PB0	PB2	INT2	Additional External Interrupt
Digital 9	PWM1	PB1	PB3	PWM0	
Digital 10	SS / PWM1	PB2	PB4	SS / PWM0	SPI
Digital 11	MOSI / PWM2	PB3	PB5	MOSI	SPI
Digital 12	MISO	PB4	PB6	MISO	SPI
Digital 13	SCK	PB5	PB7	SCK	SPI
			PB0		uSD Card - SPI SS & Through Hole (**SEE ERRATA**)
			PB1		uSD Card - Card Sense & Through Hole
SCL			PC0	SCL	Separate I2C
SDA			PC1	SDA	Separate I2C
			PC2	TCK	JTAG port
			PC3	TMS	JTAG port
			PC4	TDO	JTAG port
			PC5	TDI	JTAG port
	XTAL1	PB6	PC6	TOSC1	32.768 kHz RTC Crystal
	XTAL2	PB7	PC7	TOSC2	32.768 kHz RTC Crystal

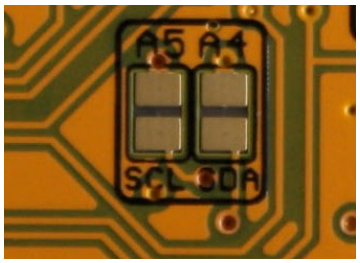
Solder Bridge Descriptions

Goldilocks has a number of solder bridge options that are designed to significantly increase the flexibility of the PCB and the use of the two available MCU, and to improve compatibility with existing Arduino shields.

I2C Arduino Pre-R3 Support

OPEN solder bridges on the rear of the PCB are used to connect the ATmega1284p PC0/PC1 to PA5/PA4 respectively. This for compatibility with Arduino Uno R3, which does not link the I2C SDA/SCL pins with the Analogue 4 and 5 pins. Older (Pre-R3) shields that have I2C pins on Analogue 4 and 5 are supported by closing the solder bridges.

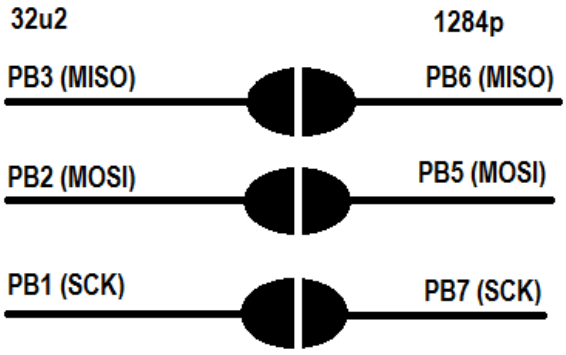


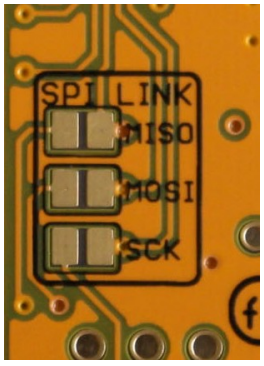


SPI Interconnect

The SCK, MISO, MOSI pins on the rear of the PCB are connectable with OPEN solder bridges on the rear of the PCB, to connect the ATmega1284p with the ATmega32U2.

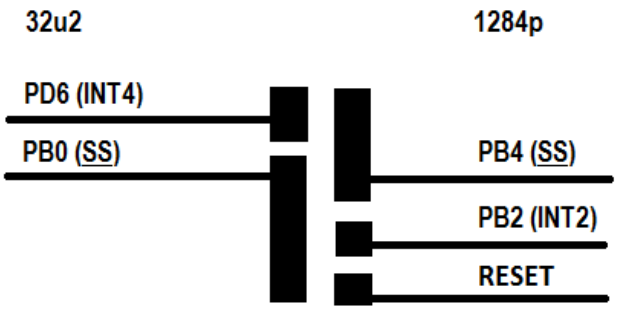
Atmel documents AVR151 and AVR910 refer to 1 k ohm to 10 k ohm in-series resistors to prevent SPI bus lock-up (but parasitic capacitance effects can limit throughput), to allow an ICSP to function despite the device being in circuit whilst being programmed. Goldilocks is implemented with 1 kOhm resistors in series with the SPI lines to provide this decoupling.

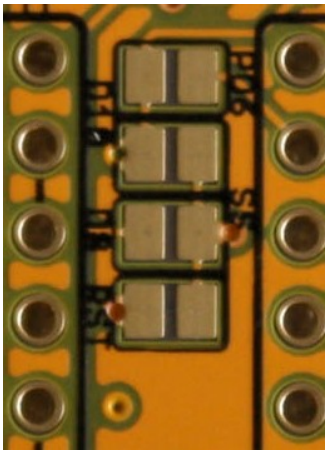




The ATmega32u2 PB0 (SS) + PD6 (INT4) pins and the ATmega1284p PB4 (SS) + PB2 (INT2) pins are brought to a row of solder pads on rear of board. The SS lines are aligned to be directly connected. Or alternatively, SS to INTx (or left unconnected) respectively if direct access to either

SPI Master SS is not desired, but Interrupt signalling the SPI Master is desired. Also, ATmega1284p RESET is provided to allow the optional use of LUFA AVR-ISP software.

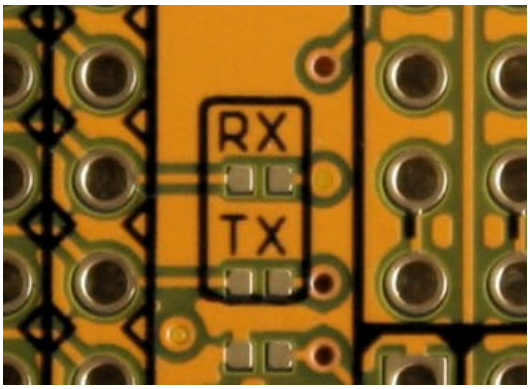




Rx Tx Disconnect

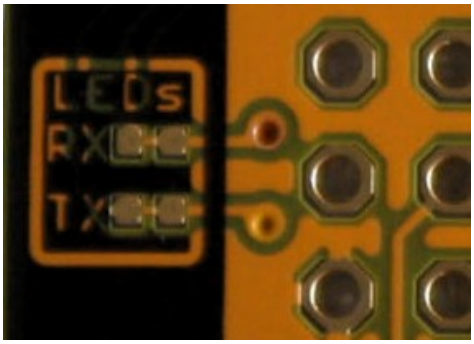
The Rx Tx Disconnect option is provided for occasions where the user wishes to have all three USART interfaces, on both ATmega1284p and ATmega32U2, available separately. This might be useful where one MCU is controlling a serial interface radio (Zigbee or otherwise), and the other MCU is driving sensors or actuators. Communication between the two MCU can then be done via the SPI bridge, noted above.

The Rx Tx bridge is located on the front of the Goldilocks and is normally equipped with resistors. These resistors need to be REMOVED to disconnect the USART interfaces on each MCU.



ATmega32U2 Rx Tx LED Disconnect

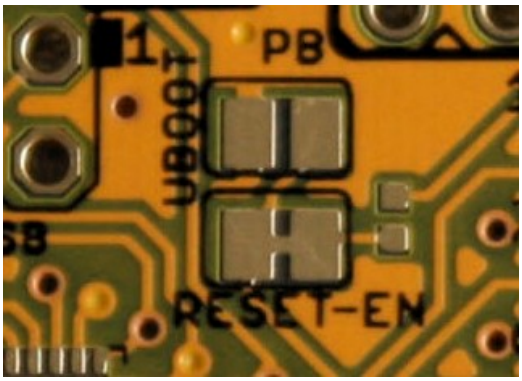
Where the Atmega32U2 is being used as an ancillary MCU and the Digital I/O is in action, it may be necessary to disable the Rx and Tx LEDs if they cause loading on the lines. The Rx Tx disconnect option is located on the rear of the Goldilocks, and is provided to enable a reversible option for this situation.



ATmega32U2 HWB (DFU Bootloader) & ATmega1284p RESET Disconnect

These Arduino standard jumpers on the front of the PCB provide the functionality to put the ATmega32U2 into its DFU bootloader state, to enable programming of the ATmega32U2.

The RESET-EN separates the ATmega1284p from control of the ATmega32U2. This is useful in the situation where it is NOT desirable that re-plugging the USB port of the Goldilocks should RESET the ATmega1284p. When this RESET-EN solder bridge is separated, it is necessary to press the ATmega1284p RESET button each time it is desired to enter its Wiring (stk500v2) bootloader for reprogramming.



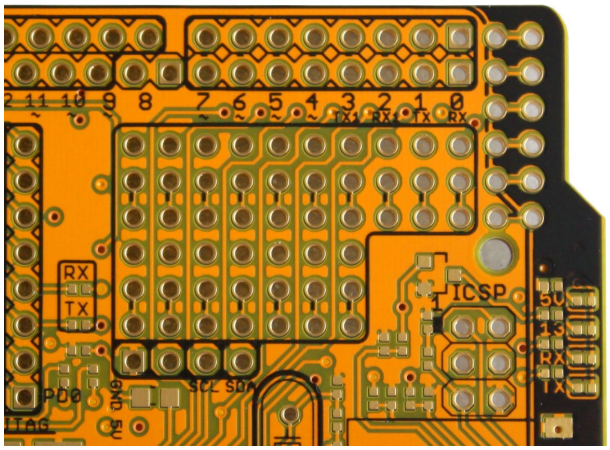
Selected Feature Descriptions

Prototyping space

Prototyping space is provided in one extended section in the upper right of the Goldilocks PCB.

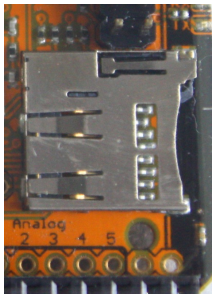
On the upper right hand corner a stitched dual row of pins is provided to allow vertically orientated modules to be mounted outside the profile of a standard Arduino Uno sized shield. This space can also be used for conveniently taking signals off the PCB without impacting shield usage.

In the centre section six rows of holes are alternately stitched to allow components or small modules to be mounted onto the PCB. Specifically I2C prototyping modules are supported by the provision of the GND, 5V, SCL, and SDA pins along the lower edge of the space.



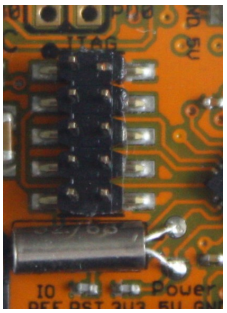
uSD Card

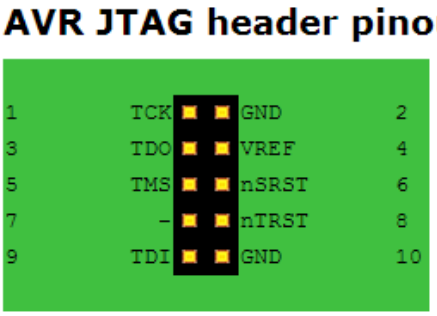
The micro SD card cage is provided with 3.3V power, and level shifted signals to properly support uSD Cards. The SPI Slave Select line is connected to PB0, which is not allocated to the standard Arduino pinouts, to ensure that full shield flexibility is not compromised. The Card Sense line is connected to PB1. If the standard Arduino SD card library or sketches are used, then the SS pin assignment must be changed to Arduino Digital 14. **PLEASE REFER TO ERRATA**

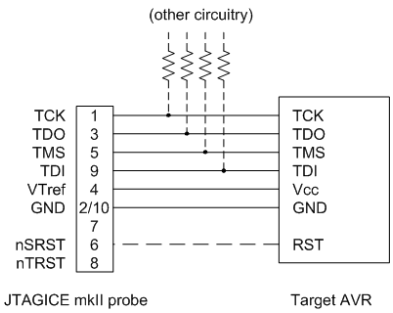


ATmega1284p JTAG Header

The IEEE Standard 1149.1 compliant JTAG header is populated according to the AVR recommendations. JTAG is not enabled by default in the AVR fuse configuration for Goldilocks.







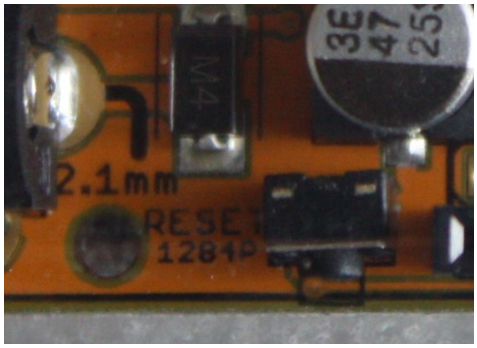
32.768kHz Crystal

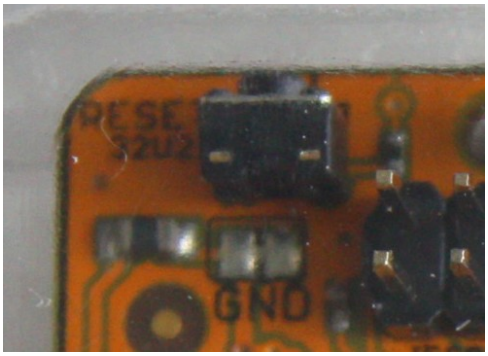
The ATmega1284p supports Timer2 being driven by an external 32.768kHz watch crystal. Using

this Timer, together with the avr-libc provided functions in the time.h library allows the creation of an accurate real-time clock capability for the Goldilocks PCB.

RESET buttons

Two separate RESET buttons are provided for the Goldilocks.
The ATmega1284p RESET, located on the lower edge of the PCB, is the “normal” Arduino RESET, and is connected to the RESET pin-out.

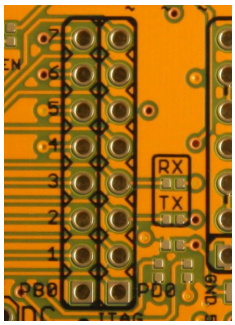




The ATmega32U2 RESET, located on the upper left corner of the PCB, enables the Goldilocks to reset its serial interface without power cycling, or resetting, the ATmega1284p.

32U2 Port Header

To make the ATmega32U2 more useful as a “coprocessor” to the ATmega1284p, two complete port pinouts, PB and PD, are provided. These two ports make the SPI bus, the USART, and external interrupt pins available.



The ATmega32U2, having 32 kBytes of Flash and 1 kByte SRAM, is nearly as resource rich as the Arduino Uno’s ATmega328p or the Leonardo’s ATmega32U4. It will be possible to develop multi-MCU solutions based on the Goldilocks platform, that use the ATmega32U2 to drive a serial device (radio, gps, or gsm, for example), whilst the ATmega1284p manages a complex analogue data acquisition or communications task for example.

AVR Fuses

The fuses are set slightly differently to the normal Arduino practice. This describes the actual settings on the Goldilocks PCB, for each of the MCU.

Both the ATmega1284p and the ATmega32U2 are being operated at their maximum specified system clock. The clock divide by 8, CKDIV8, on both MCU is set to off, so that they can operate at

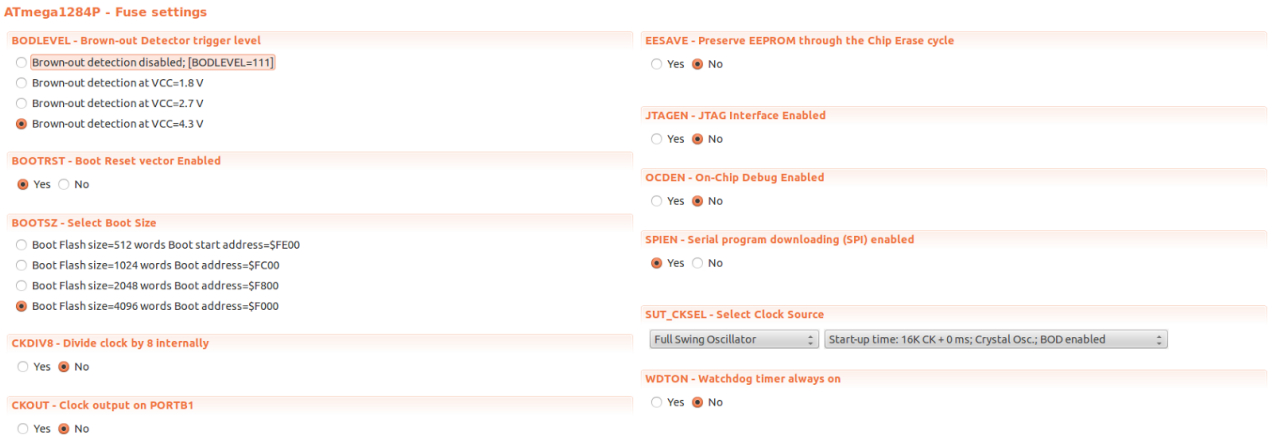
this rate.

Because of the system clock rate, both MCU require greater than 4.3V for reliable operation. To ensure that the MCU stay in RESET state until their Vcc is adequate the brown-out detection level, BODLEVEL, on both MCU is set to 4.3V.

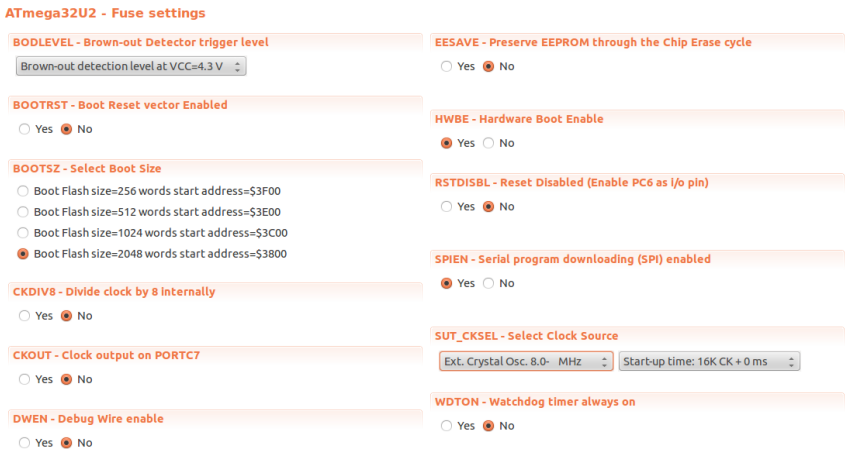
Setting the BODLEVEL correctly ensures that the MCU will start immediately, once it is released from RESET. This allows us to reduce the startup time for each MCU to 16k CLK without risk of instability, or need for further delay. Both MCU are driven from Full Swing Oscillators, to further ensure stable operation, as specified by their respective data-sheets.

In both cases the EESAVE is set to clear the EEPROM each time the MCU is erased, and the system watchdog timer is turned off by setting WDTON off.

For the ATmega1284p, the boot reset, BOOTRST, vector is enabled, as we would like the Wiring (stk500v2) bootloader to respond once the MCU is reset. But, both JTAG enable, JTAGEN, and on-chip debug, OCDEN, are turned off, as they are advanced features that should be enabled on a case by case basis.

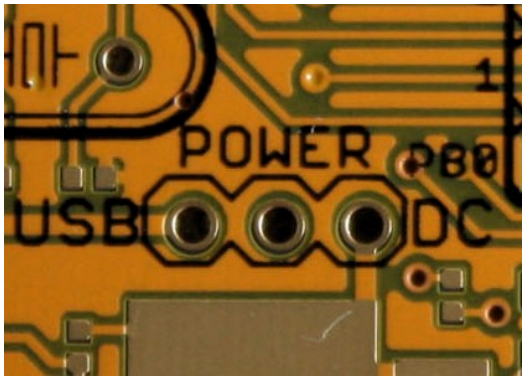

ATmega1284P - Fuse settings
BODLEVEL - Brown-out Detector trigger level
Brown-out detection disabled; [BODLEVEL=111]
Brown-out detection at VCC=1.8 V
Brown-out detection at VCC=2.7 V
Brown-out detection at VCC=4.3 V
BOOTRST - Boot Reset vector Enabled
Yes No
BOOTSZ - Select Boot Size
Boot Flash size=512 words Boot start address=$FE00
Boot Flash size=1024 words Boot address=$FC00
Boot Flash size=2048 words Boot address=$F800
Boot Flash size=4096 words Boot address=$F000
CKDIV8 - Divide clock by 8 internally
Yes No
CKOUT - Clock output on PORTB1
Yes No
EESAVE - Preserve EEPROM through the Chip Erase cycle
Yes No
JTAGEN - JTAG Interface Enabled
Yes No
OCDEN - On-Chip Debug Enabled
Yes No
SPIEN - Serial program downloading (SPI) enabled
Yes No
SUT_CKSEL - Select Clock Source
Full Swing Oscillator
Start-up time: 16K CK + 0 ms; Crystal Osc.; BOD enabled
WDTON - Watchdog timer always on
Yes No

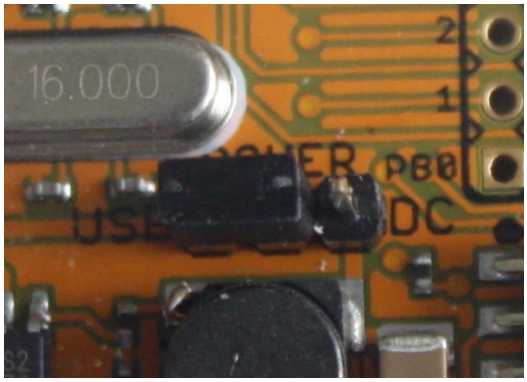
For the ATmega32U2, the BOOTRST vector is disabled, as we don’t need the MCU to enter the DFU bootloader, unless we activate the hardware boot pin. This hardware boot enable is set by the hardware boot enable, HWBE, fuse.


ATmega32U2 - Fuse settings
BODLEVEL - Brown-out Detector trigger level
Brown-out detection level at VCC=4.3 V
BOOTRST - Boot Reset vector Enabled
Yes
No
BOOTSZ - Select Boot Size
Boot Flash size=256 words start address=$3F00
Boot Flash size=512 words start address=$3E00
Boot Flash size=1024 words start address=$3C00
Boot Flash size=2048 words start address=$3800
CKDIV8 - Divide clock by 8 internally
Yes
No
CKOUT - Clock output on PORTC7
Yes
No
DWEN - Debug Wire enable
Yes
No
EESAVE - Preserve EEPROM through the Chip Erase cycle
Yes
No
HWBE - Hardware Boot Enable
Yes
No
RSTDISBL - Reset Disabled (Enable PC6 as I/o pin)
Yes
No
SPIEN - Serial program downloading (SPI) enabled
Yes
No
SUT_CKSEL - Select Clock Source
Ext. Crystal Osc. 8.0- MHz
Start-up time: 16K CK + 0 ms
WDTON - Watchdog timer always on
Yes
No

Power Configuration

Goldilocks can be powered by a variety of options, to provide a 5V supply to the main power. Additionally a 3.3V regulator provides supply for the on-board uSD card and for external shield usage.





By default the Goldilocks is powered via 5V USB, with the maximum input current limited to 500mA.

By moving the POWER jumper to DC the Goldilocks can be powered from a high current 5V 2A Switch Mode Power Supply, via the 2.1mm (+)ve centre barrel connector.

Another option is to provide 5V directly to the Vcc supply rail. Care must be taken not to exceed the maximum voltage rating of the ATmega1284p and ATmega32U2 in this case, otherwise the Goldilocks PCB will be dead.

Because both the ATmega1284p and the ATmega32U2 are operated at the maximum system clock frequency specified, their respective brown-out fuses are set to ensure that the MCU will not operate until a sufficient supply voltage (4.3V) is available.

Arduino Compatibility

The Goldilocks was designed to be fully compatible with the Arduino R3 standard. In the last year, with the release of ARM platform Arduino boards, the meaning of being fully compatible has become a little greyer and more complex. Hopefully, the Goldilocks reaches the level where it can be considered fully compatible, and possibly being even more “compatible”, with the Arduino legacy, than the Arduino Due.

Features

The standard Arduino IDE used. The Arduino IDE v1.0.5 was used in development. Prior versions of the IDE provide older versions of the avr-libc that contains some significant bugs.

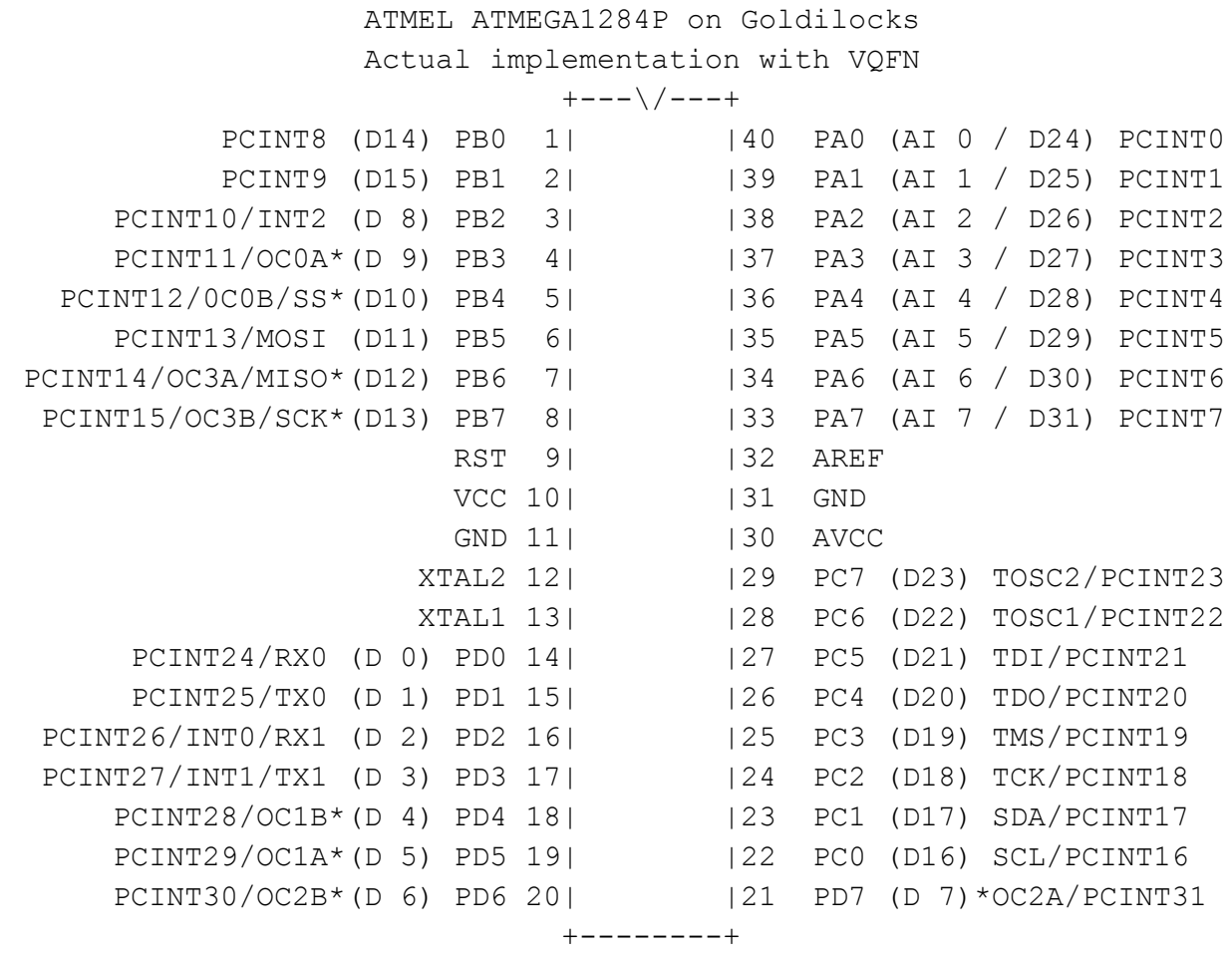
Compatible bootloaders are used in both ATmega1284p and ATmega32U2. Compatible pin mappings are used to ensure that Arduino Uno R3 shields should work as intended, and with some adjustments to the solder jumpers, Pre-R3 Uno shields will work too.

Software development was done on using avr-gcc (GCC) 4.7.2-2 & avr-libc 1.8.0-3 from Debian Sid repositories on a Ubuntu 12.04 platform.

If you are motivated, get a modern avr-gcc compiler, such as the gcc 4.7.x branch. The avr-gcc 4.3.2 version, included with Arduino IDE v1.0.5 and earlier, is over 5 years old and contains many issues. Support for the ATmegaxxU2 was only added in gcc 4.5.x, for example.

Details

Over the past years, much work in the core of the avr-libc has been done to support the ATmega1284p. Also, within the Arduino community there has been a push to incorporate the ATmega1284p into standard cores. This means that the Goldilocks can be operated in the Arduino IDE environment with minimal additional information. Purely the boards.txt file is needed to configure the IDE, and also then pins_arduino.h file contained in the variants/goldilocks directory is needed to make sure that the pins are mapped to known Arduino standards.


ATMEL ATMEGA1284P on Goldilocks
Actual implementation with VQFN
PCINT8 (D14) PB0 1
PCINT9 (D15) PB1 2
PCINT10/INT2 (D 8) PB2 3
PCINT11/OC0A*(D 9) PB3 4
PCINT12/0C0B/SS*(D10) PB4 5
PCINT13/MOSI (D11) PB5 6
PCINT14/OC3A/MISO*(D12) PB6 7
PCINT15/OC3B/SCK*(D13) PB7 8
RST 9
VCC 10
GND 11
XTAL2 12
XTAL1 13
PCINT24/RX0 (D 0) PD0 14
PCINT25/TX0 (D 1) PD1 15
PCINT26/INT0/RX1 (D 2) PD2 16
PCINT27/INT1/TX1 (D 3) PD3 17
PCINT28/OC1B*(D 4) PD4 18
PCINT29/OC1A*(D 5) PD5 19
PCINT30/OC2B*(D 6) PD6 20
40 PA0 (AI 0 / D24) PCINT0
39 PA1 (AI 1 / D25) PCINT1
38 PA2 (AI 2 / D26) PCINT2
37 PA3 (AI 3 / D27) PCINT3
36 PA4 (AI 4 / D28) PCINT4
35 PA5 (AI 5 / D29) PCINT5
34 PA6 (AI 6 / D30) PCINT6
33 PA7 (AI 7 / D31) PCINT7
32 AREF
31 GND
30 AVCC
29 PC7 (D23) TOSC2/PCINT23
28 PC6 (D22) TOSC1/PCINT22
27 PC5 (D21) TDI/PCINT21
26 PC4 (D20) TDO/PCINT20
25 PC3 (D19) TMS/PCINT19
24 PC2 (D18) TCK/PCINT18
23 PC1 (D17) SDA/PCINT17
22 PC0 (D16) SCL/PCINT16
21 PD7 (D 7)*OC2A/PCINT31

* = PWM capable pin
TOSCn = RTC Crystal pinout (not otherwise available)
TCK/TMS/TDO/TDI = JTAG pinout (not otherwise available)

```
##############################################################
#
# ADD THESE ENTRIES TO YOUR FILE BOARDS.TXT
#
##############################################################

goldilocks_20MHz.name=Goldilocks 20MHz

goldilocks_20MHz.upload.protocol=wiring
goldilocks_20MHz.upload.maximum_size=122878
goldilocks_20MHz.upload.speed=38400

goldilocks_20MHz.bootloader.low_fuses=0xd7
goldilocks_20MHz.bootloader.high_fuses=0xd8
goldilocks_20MHz.bootloader.extended_fuses=0xfc
goldilocks_20MHz.bootloader.path=stk500v2
goldilocks_20MHz.bootloader.file=stk500boot_v2_goldilocks.hex
goldilocks_20MHz.bootloader.unlock_bits=0x3F
goldilocks_20MHz.bootloader.lock_bits=0x0F

goldilocks_20MHz.build.mcu=atmega1284p
goldilocks_20MHz.build.f_cpu=20000000L
goldilocks_20MHz.build.core=arduino
goldilocks_20MHz.build.variant=goldilocks
##############################################################

goldilocks_22MHz.name=Goldilocks 22.1184MHz

goldilocks_22MHz.upload.protocol=wiring
goldilocks_22MHz.upload.maximum_size=122878
goldilocks_22MHz.upload.speed=38400

goldilocks_22MHz.bootloader.low_fuses=0xd7
goldilocks_22MHz.bootloader.high_fuses=0xd8
goldilocks_22MHz.bootloader.extended_fuses=0xfc
goldilocks_22MHz.bootloader.path=stk500v2
goldilocks_22MHz.bootloader.file=stk500boot_v2_goldilocks_22.hex
goldilocks_22MHz.bootloader.unlock_bits=0x3F
goldilocks_22MHz.bootloader.lock_bits=0x0F

goldilocks_22MHz.build.mcu=atmega1284p
goldilocks_22MHz.build.f_cpu=22118400L
goldilocks_22MHz.build.core=arduino
goldilocks_22MHz.build.variant=goldilocks
```

Known Arduino R3 Incompatibilities

Mounting holes are set to be 2mm in diameter (M2). The Arduino standard is 3mm (M3). The hole size needed to be reduced to allow Port A pins to be continuous.

PWM supported pins on the ATmega1284p are different to the Arduino Uno ATmega328p. This means that shields requiring the use of PWM (such as motor H-bridge shields) may need to have the pins rerouted either on the Goldilocks, using the provided pin-out options, or on the shield itself. This issue also affects the Arduino Leonardo ATmega32U4.

The actual PWM support is better than in the Arduino Uno as the important 16 bit Timer pins, that can generate correct servo timing pulses, are not located on the SPI pins. Also the ATmega1284p has an extra 16bit timer, Timer3, that can also be used for PWM servo control too, allowing 4 servos to be directly controlled.

USART serial port speed accuracy and mismatch remains a concern. An acceptable compromise has been provided by implementing the correct calculation for serial baud rate contained in avr-libc util/setbaud.h. However, the total error in serial rate between the AVR MCU at 16 MHz (ATmega32U2) and at 20 MHz (ATmega1284p) unfortunately remains quite high. The Wormfood AVR Serial Rate Calculator indicates the issue with getting an accurate baud rate from a 16 MHz ATmega MCU.

If serial communications proves an issue for you, the easy answer is to reduce the ATmega1284p USART to 38,400 baud which provides the lowest differential error (at a reasonably fast communication rate). The ATmega32U2 will auto configure to match the baud rate of your attached terminal programme. Alternatively, the best solution is to operate the ATmega1284p at a USART friendly system clock of 22.1184 MHz, 18.432 MHz, or lower.

Because of the issues with serial communications, and the inability of the 16 MHz ATmega32U2 to generate an accurate USART baud rate at either 115,200 or 57,600, the stk500v2 bootloader on the ATmega1284p has been set to operate at 38,400 baud. This does not reduce the actual programming speed for sketches or programmes, but it does allow the total mismatch in baud rates between the two MCU to remain less than 1%.

ATmega1284p Firmware

Wiring (stk500v2) ATmega1284p Bootloader

To access the ATmega1284p bootloader monitor open up a serial port at 38,400 baud and type "!!!" within 3 seconds of RESET. The ATmega1284p then enters the stk500v2 "Bootloader>" prompt, which is a useful sekret feature.

If you want to see what is actually programmed into the Goldilocks, then this feature allows you to examine the Flash, and EEPROM. It also allows you to see which Vectors have been programmed and to test the hardware by blinking ports.

The bootloader commands are:

- 0 Reset address (RAM, EEPROM, FLASH) counters to zero.
- ? CPU stats
- @ EEPROM test
- b Blink LED
- e Dump EEPROM
- f Dump FLASH
- h Help
- l List available I/O Ports
- q Quit and jump to user program
- r Dump RAM
- v show interrupt Vectors
- y Port blink

ATmega32U2 Firmware

The ATmega32U2 firmware was extracted directly from a snapshot from the 130313 release of the LUFA repository. The snapshot was taken on May 20th, but it differs negligibly from the standard release.

The choice to use standard LUFA code (with a minimum of divergence) was taken to ensure that the Goldilocks platform would continue to benefit from Dean Camera's code development efforts and bug fixes. The current Arduino releases of xxU2 code were forked from LUFA in 2010, and haven't been significantly maintained since then.

LUFA DFU Bootloader

The DFU bootloader used in the ATmega32U2 is unchanged from the standard maintained by LUFA. Information relating to this code can be found on the LUFA repository.

LUFA Benito (U2duino) Serial Interface

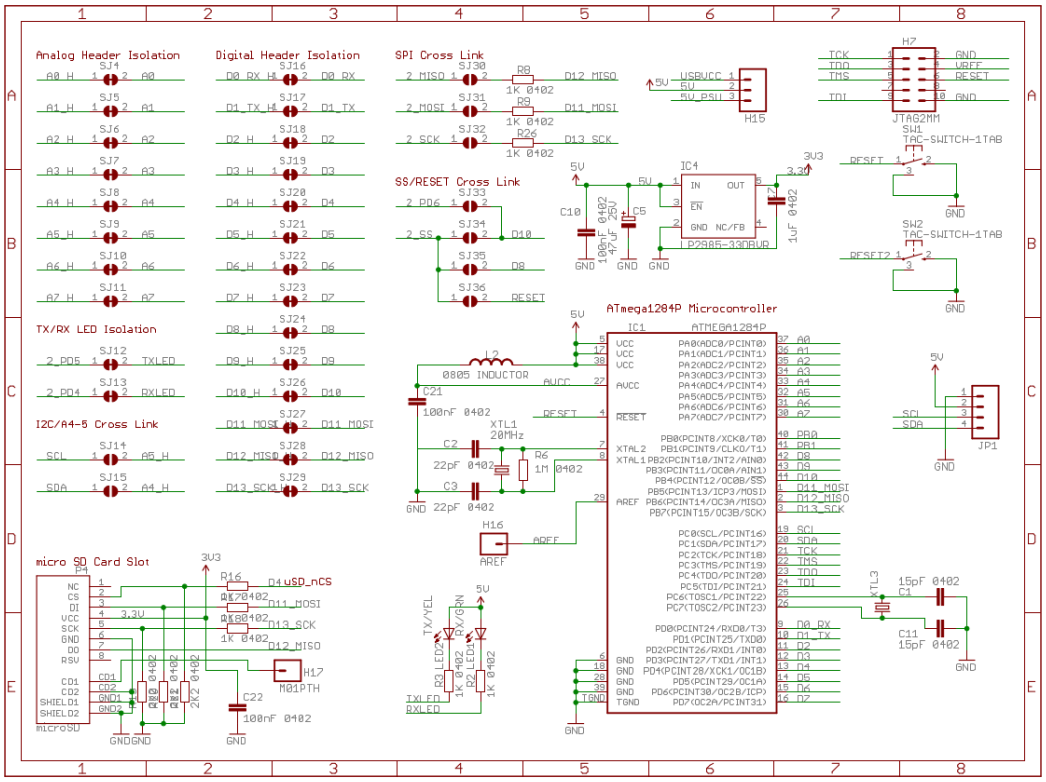
The LUFA project Benito was used as the basis for the USB to Serial interface for the Goldilocks. The code has been renamed U2duino, and slightly modified to support the wiring or arduino bootloader DTR reset functionality.

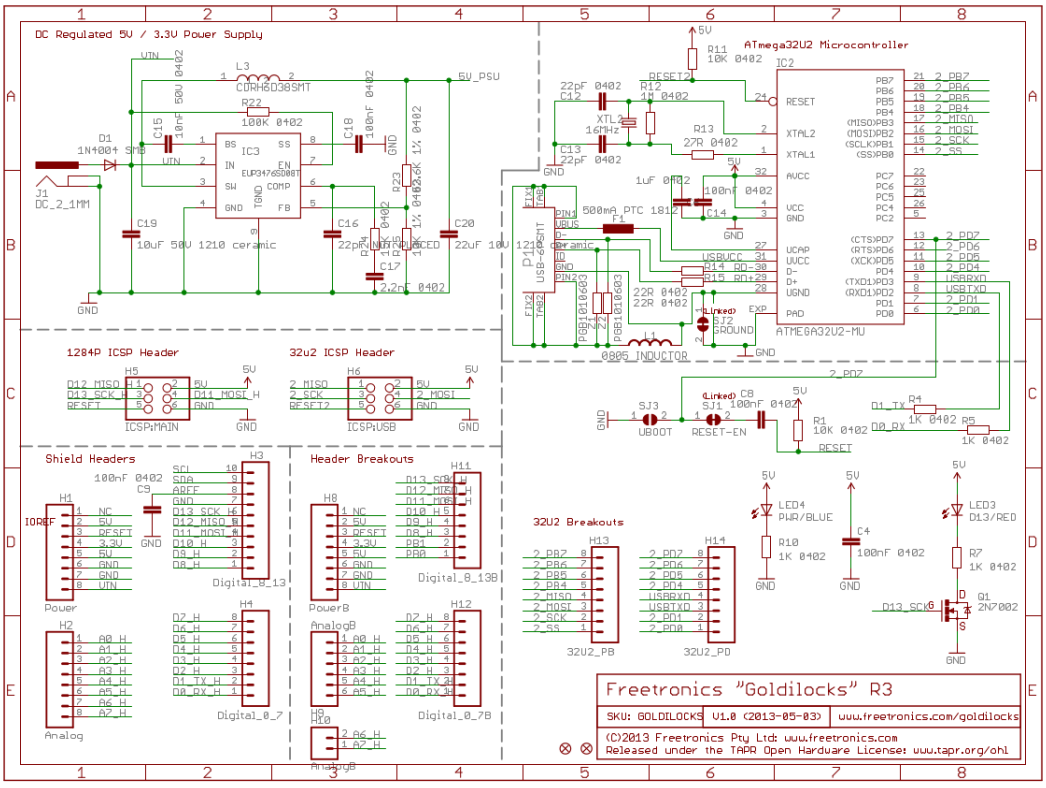
The project name U2duino was chosen because the code is for xxU2 on the xxduino board, or alternatively the project does: USB-2-Arduino.

The key difference from Benito is the pin driving the ATmega1284p RESET line is toggled OUT_HIGH to OUT_LOW (rather than TRISTATE to OUT_LOW). This ensures that a clear negative edge is generated (via a serial capacitor) on the DTR signal, which properly resets the ATmega1284p.

Another key difference is to modify the calculation for USART baud rate to be the same as in the avr-libc file util/set_baud.h file. This calculation method is more correct than the calculation method used in the AVR datasheet.

Schematic


Analog Header Isolation
Digital Header Isolation
SPI Cross Link
SS/RESET Cross Link
TX/RX LED Isolation
I2C/A4-5 Cross Link
ATmega1284P Microcontroller
micro SD Card Slot
JTAG2MM
TAC-SWITCH-1TAB
P2985-330RUB
0805 INDUCTOR
XTL1 20MHz
XTL3 16MHz
JP1


DC Regulated 5V / 3.3V Power Supply
ATmega32U2 Microcontroller
1284P ICSP Header
32u2 ICSP Header
Shield Headers
Header Breakouts
32U2 Breakouts
ATMEGA32U2-MU
Freetronics "Goldilocks" R3
SKU: GOLDILOCKS V1.0 (2013-05-03) www.freetronics.com/goldilocks
(C)2013 Freetronics Pty Ltd: www.freetronics.com
Released under the TAPR Open Hardware License: www.tapr.org/ohl

Errata

- ATmega1284p - if the serial port Tx in use, the serial bootloader won't function, because avrdude stk500v2 protocol doesn't see the response bytes it expects, and you may be using it at a different rate. An inherent serial bootloader issue.
- ATmega32u2 - used Atmel USB descriptors (rather than Freetronics). But Atmel descriptors have a standard CDC INF file for Windows.
- The uSD card frame Slave Select SS, is not connected to PB0 (Digital 14) as per design, but rather to the normal Arduino Digital 4 (PD4).This impacts the use of 16bit PWM (found on PD4 and PD5) when the SD card is in use. This also impacts the use of multiple SD cards, as the Goldilocks SD card SS line is now shared with the standard Arduino layout.
- The uSD card detect line is connected to a through hole close to the rear of the uSD card cage. It is not otherwise connected to any pin, as it is not normally used. If it is required, then jumper from the through hole to a digital pin of your choice.
- The JTAG VREF pin is not connected. It should nominally be connected to the MCU supply reference voltage of 5V. This should not affect the use of the JTAG connector.

Support

Contact Information

I can be contacted via:
twitter @fei_li_pu
blog feilipu.me
sourceforge https://sourceforge.net/projects/avrfreertos/

Support Information

If you have received a Goldilocks because you made pledge on Pozible, please contact me as above, or via a message on the Goldilocks Pozible Project.

If you purchased a Goldilocks from Freetronics, or from any distributor reselling the Goldilocks, please contact Freetronics via their support forum.

Precautions

WARNING
Care must be taken when configuring the solder jumpers. It is possible to cause permanent damage to the device or the power supply by improperly setting the jumpers. Do not change any jumpers while the unit is powered. When using the on-board SMPS regulator to power the Goldilocks, be sure not to exceed the specified maximum current output.

CAUTION
The Goldilocks PCB contains static sensitive components. Use the usual ESD procedures when handling.

CAUTION
Improper fuse settings may result in an unusable Goldilocks PCB. Be certain that you know the effects of changing the fuses, that you understand the convention used for describing the state of the fuses (programmed = 0), and that you are using an appropriate programming speed before attempting to change fuse settings.

Legal

Copyright Notices

This code uses the LUFA USB library Copyright (C) 2013, Dean Camera (www.fourwalledcubicle.com) and distributed under a modified MIT license (see files).
The Benito Project and DFU bootloader are modified versions from LUFA 130313.
Portions of this code are copyright © 2003-2013, Atmel Corporation (http://www.atmel.com/)

Software Warranty Disclaimer

The author disclaim all warranties with regard to this software, including all implied warranties of merchantability and fitness. In no event shall the author be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of this software.

Hardware Disclaimer

This development board is intended for use for FURTHER ENGINEERING, DEVELOPMENT, DEMONSTRATION, OR EVALUATION PURPOSES ONLY. It is not a finished product, and may not (yet) comply with some or any technical or legal requirements that are applicable to finished products, including, without limitation, directives regarding electromagnetic compatibility, recycling (WEEE), FCC, CE, or UL (except as may be otherwise noted on the board).

I, Phillip Stevens, have supplied this board AS IS, without any warranties, with all faults, at the buyer's and further users' sole risk. The user assumes all responsibility and liability for proper and safe handling of the goods. Further, the user indemnifies me, Phillip Stevens, from all claims arising from the handling or use of the goods. Due to the open construction of the product, it is the user's responsibility to take any and all appropriate precautions with regard to electrostatic

discharge and any other technical or legal concerns.

The product described in this document is subject to continuous development and improvements. All particulars of the product and its use contained in this document are given by me in good faith. However all warranties implied or expressed including but not limited to implied warranties of merchantability or fitness for particular purpose are excluded.

This document is intended only to assist the reader in the use of the product. I, Phillip Stevens, shall not be liable for any loss or damage arising from the use of any information in this document or any error or omission in such information or any incorrect use of the product.

Trademarks

AVR® is a registered trademark of Atmel Corporation.
All other trademarks are the property of their respective owners.

Acknowledgments

Thanks to Dean Camera (http://www.fourwalledcubicle.com/) for his excellent LUFA library and DFU bootloader, which are used in the Goldilocks firmware.

Thanks to Pozible for providing their crowdfunding platform.

Thanks to those who have previously used the ATmega1284p or ATmega644 in Arduino projects and have smoothed the way for incorporating my Goldilocks into the Arduino family of options.

Finally, thanks to Jon Oxer for helping me realise this Goldilocks project, and graciously accommodating my harping and feature creep over the four months of project realisation.